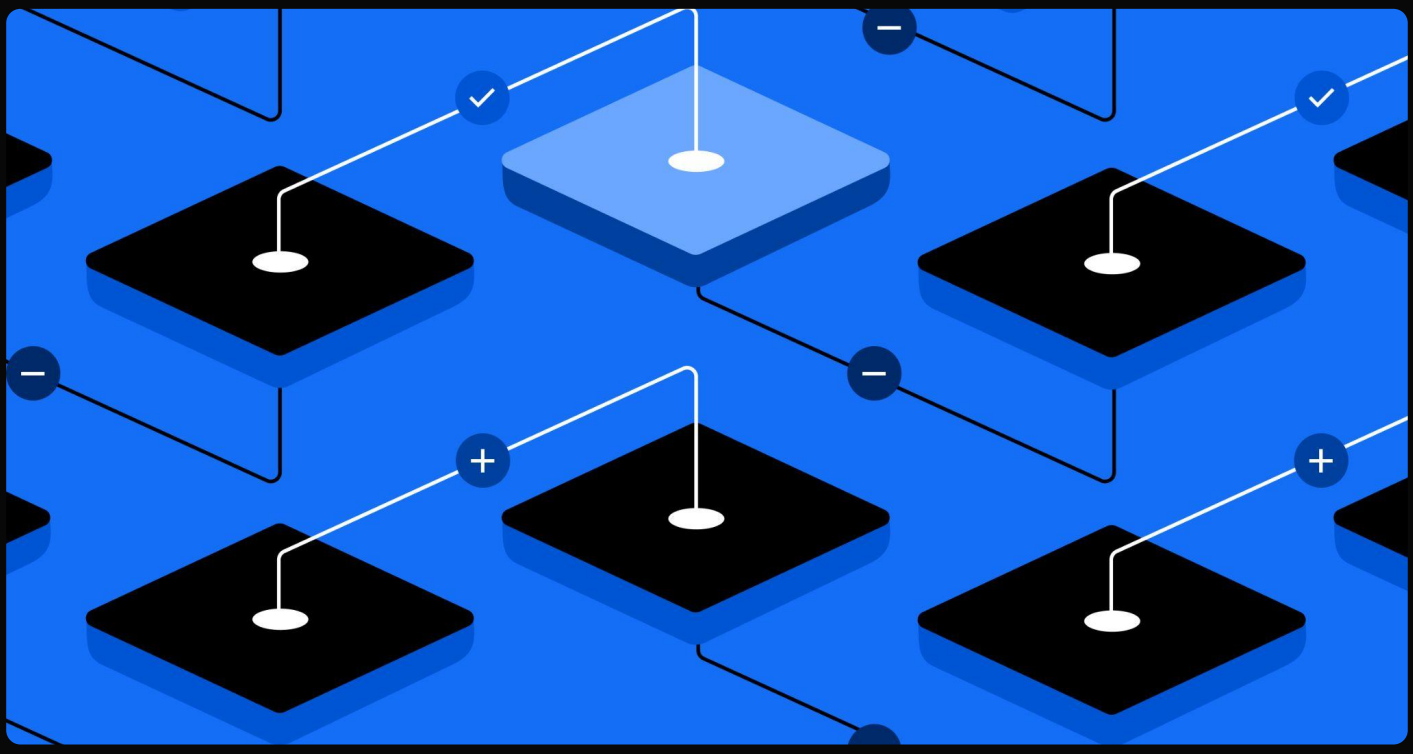
ACTIVITY GUIDE

Take your app live with ↗
Webflow Cloud

Extend & redeploy a Webflow Cloud app

Use a prepared app to practice the full Webflow Cloud workflow: create your own repository, deploy the starter, make a change, and push that change live.



OVERVIEW

What you'll build

This activity takes you through the complete workflow for deploying and extending a Webflow Cloud app. You'll begin with a working starter, improve how it displays shared results, and push your update live through GitHub.

After initial setup, you'll complete the activity in two stages.

1

Visualize the results

Add percentage bars and visible text that identifies your selection and the leading result.

2

Redeploy your change

Test the update locally, commit it to GitHub, and verify the new Webflow Cloud deployment.

Before you start, you'll need:

- [ ] A GitHub account
- [ ] Access to Webflow Cloud
- [ ] Git and Node.js 22.12 or later
- [ ] A terminal coding agent, such as Claude Code or Codex CLI

Note — The template already includes the app code, deployment configuration, database, and sample responses. You won't be starting from scratch.

If you need extra guidance, try providing this entire document to an AI agent and asking it to walk you through the process, step by step.

SETUP · STEPS 1 – 2

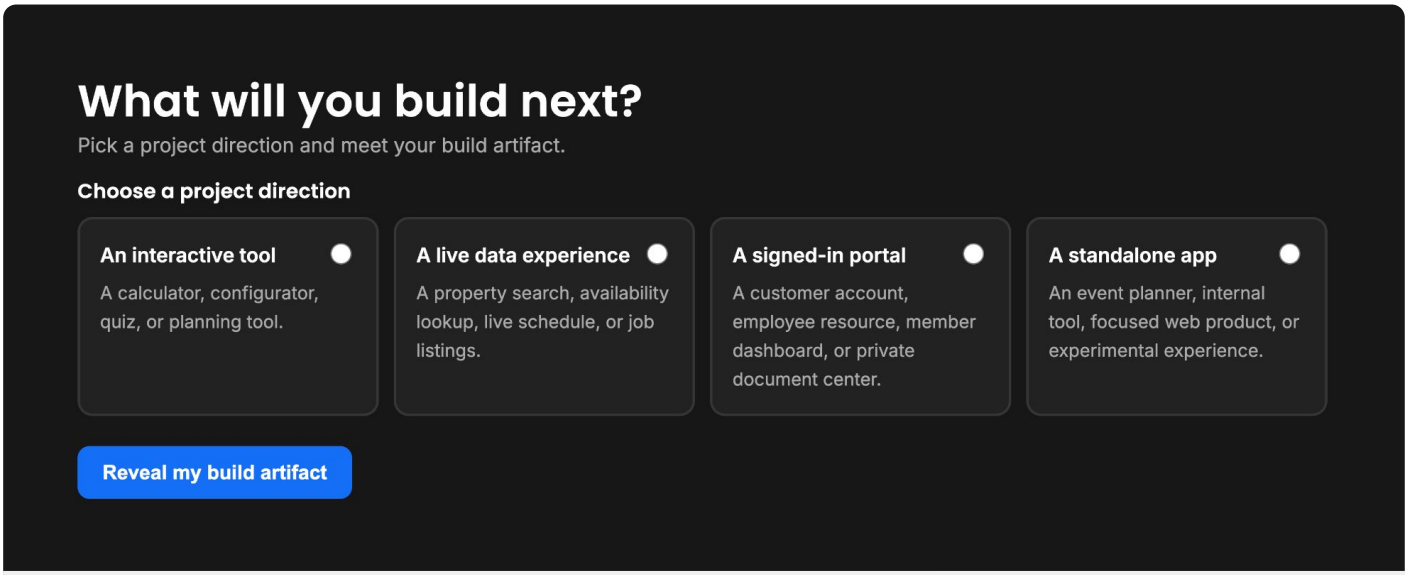
Create & deploy the starter

1. Create your GitHub repository

- Open the **What will you build next?** template on GitHub, select **Use this template**, then **Create a new repository**.
- Choose your account or organization as the owner, give the repository a name you'll recognize, and select the visibility that fits your work.
- Select **Create repository from template**. This creates your own copy of the starter app, ready to deploy to Webflow Cloud.

2. Deploy the starter in Webflow Cloud

- Log in to Webflow, open your Workspace, and select **New project** → **App**.
- Authorize GitHub access if prompted, expand **Import a GitHub repository**, and import your new repo.
- Name the app, select the main branch, and select **Deploy**.
- When the deployment succeeds, open the app and complete a few responses. Confirm that the shared result count updates as you submit responses.



View of the deployed app

Open the project

3. Clone the repository locally

- Open your new repository on GitHub. Select **Code**, make sure **HTTPS** is selected, and copy the repository URL.
- Open your terminal and move to the folder where you want to keep the project. Then run:

```
git clone <repository-url>
cd <repository-name>
```

Repository root means the project's main folder. You're in the right place when the folder contains files such as `README.md` and `package.json`, along with the `src` folder.

4. Install the project and start your coding agent

- From the repository root, install the project dependencies:

```
npm install
```

- Then launch your coding agent from the same folder. For **Claude Code** or **Codex CLI**:

```
claude
```

```
codex
```

- Follow any sign-in or permission prompts that appear. Because you launched the agent from the repository root, it can now inspect and update the project.

Other ways to work — If you use another terminal coding agent, launch it from the repository root using its usual command. GitHub Codespaces and manual editing also work; see the `README.md` for alternate setup and troubleshooting.

ACTIVITY · PROMPT

Visualize the results

Next, you'll turn the text-only results list into an easy-to-scan visualization. Paste the prompt below into the coding agent you launched from the repository root.

The finished result should:

- Add percentage bars using each choice's existing accent color.
- Clearly identify your choice and preserve the existing leader and tied-leader labels.
- Remain understandable without relying on color alone.

COPY THIS PROMPT

```
Help me complete the activity in this repository.

Read the README, then inspect src/pages/index.astro. Find ACTIVITY STARTER
and the buildResultRow() function.

Modify only src/pages/index.astro. Extend buildResultRow() so each result
includes a horizontal percentage bar using the choice's accent color. Add
visible "— your pick" text to the selected choice while preserving the
existing sole-leader and tied-leader labels and data attributes.

Use the CSS in learner-resources/results-styles.css. Paste those rules
into the designated <style is:global> block near the bottom of
index.astro.

Do not change the database, migrations, API routes, choice definitions, or
the learner resource files.

Run npm run build and git diff --check, then summarize the changes and
validation results. Do not commit or push anything.
```

Review the next page before committing the changes →

ACTIVITY · SHIP IT

Review, test, & deploy

1. Review and test locally

- Only `src/pages/index.astro` changed.
- The prepared CSS is inside the designated global style block.
- `npm run build` and `git diff --check` passed.

Ask your agent:

```
Start the local development server and tell me which URL to open.
```

Open the URL in your browser, submit a choice, and verify that the horizontal bars appear:



2. Commit and deploy

```
git status
git add src/pages/index.astro
git commit -m "Visualize shared results"
git push origin main
```

Open Webflow Cloud and watch for the new deployment. When it finishes, open the live app and repeat the same checks.

FINISH AND VERIFY

Keep building

You've completed the full workflow for deploying and extending a Webflow Cloud app. Use this checklist to confirm the live result, then keep exploring if you want to make the app your own.

Completion checklist

- [ ] The starter deployed successfully before you edited it.
- [ ] The results include percentage bars and visible selection and leader labels.
- [ ] `npm run build` completes successfully.
- [ ] Your commit appears on GitHub, and Webflow Cloud deploys the latest commit from `main`.
- [ ] The deployed app records and restores a response without duplicating it.

Support

- `README.md` for setup and troubleshooting
- `learner-resources/agent-prompts.md` for the copy-ready prompt
- `learner-resources/results-styles.css` for the prepared results styles
- Webflow Cloud getting started for setup and deployment guidance
- Webflow Cloud deployments for deployment history, logs, and troubleshooting
- Take your app live with Webflow Cloud to return to the Webflow University course

Keep exploring — Ask your agent to help you refine the data visualization or try an entirely new feature. Test the change locally, then commit and push it to main to trigger another deployment in Webflow Cloud.